

LUỒNG (THREAD) (cont)

- Lợi ích của Multithreaded: mang 4 ích lợi chính:
 - **Khả năng đáp ứng:** Multithread giúp những ứng dụng tương tác với thể hoạt động thấp hơn vì ngay cả lúc 1 phần chương trình bị block hoặc cần một thời kì dài để hoạt động, chương trình nhìn chung vẫn mang thể chạy. Và điều này giúp quý khách ưng ý hơn vì khả năng đáp ứng cao của áp dụng. Điều này đặc trưng đúng sở hữu khách hàng là các designer. Bận với thể hình dung: khi 1 chương trình đang chạy, và người mua nhấn vào một nút lệnh mà cần đa số thời gian để process, thì 1 hệ thống dạng single-threaded sẽ không kích hoạt bất kì hoạt động nào khác cho tới lúc hoàn thành bước lệnh vừa rồi. Trái lại, vận dụng dạng multithread sẽ ko làm cho ngất quãng quá đa dạng quá trình của người mua vì khi mà một thread được kích hoạt để thực hành bước lệnh kia, 1 thread khác sẽ được kích hoạt để thực hành bất kì bước lệnh ít tốn thời gian hơn mà quý khách đề xuất.
 - **Khả năng san sát tài nguyên:** những tiến trình chỉ sở hữu thể san sát dữ liệu duyệt y các kĩ thuật như shared memory (vùng bộ nhớ chung) và message sharing (chia sẻ tin). Những kĩ thuật này chỉ mang thể được thiết lập bởi lập trình viên. Bên cạnh đó, những luồng san sẻ thông báo hoặc tài nguyên theo hệ thống được mặc định. Lợi ích của việc san sát code và dữ liệu là nó giúp ứng dụng có phổ biến threads hoạt động trong một vùng liên hệ chung.
 - **Tiết kiệm:** việc phân phối tài nguyên và dữ liệu cho quá trình tạo process rất tốn kém. Và vì threads tự động chia sẻ data cho process mà nó thuộc về, việc tạo những thread cho việc context-switch sẽ giúp tiết kiệm chi phí gần như. Không chỉ tầm giá mà còn là thời gian, vì việc tạo một process mới sẽ lâu hơn nhiều so sở hữu tạo 1 thread mới. Như trong Solaris, tạo ra một process lâu hơn 30 lần so mang tạo ra một thread trong process Đó, và lâu hơn 5 lần so sở hữu tạo 1 context-switch.
 - **Scalability:** lợi ích của multithreaded mô tả rõ hơn trong kiến trúc đa xử lý (multiprocessor architecture), vì multithread giúp những threads hoạt động đồng thời trong các lõi xử lý khác nhau, trong khi đối với tiến trình dạng single-threaded, 1 thread chỉ sở hữu thể chạy trên 1 bộ xử lý, không quan trọng việc có bao lắm thread trong hệ thống hiện nay.

những mô hình TRONG MULTITHREADING

Sau phần bài tổng quan về luồng - threads, bài viết này sẽ đề cập về các mô phỏng đa luồng (multithreading models). Trước nhất, threads mang thể được tương trợ qua hai phương pháp sau: duyệt các bạn, để thành lập user threads, và duyệt y nhân máy tính, để hình thành kernel threads. User threads được hỗ trợ trên kernel và được quản lý ko cần sự tương trợ trong

khoảng kernel, trong khi đó kernel threads được hỗ trợ và điều hành trực tiếp từ hệ quản lý. Các hệ điều hành hiện nay như Window, Linux, Mac OS X, Solaris đều hỗ trợ kernel threads.

Dù là hai đơn vị biệt lập, user threads và kernel threads mang mối liên hệ không thể tách rời. Ba mô phỏng sau cũng là ba cách để hình thành mối quan hệ giữa user threads và kernel threads: mô hình Many-to-one, mô hình one-to-one, và mô hình one-to-many.

Mỗi hoặc đa dạng user thread phải được map sang một hoặc phổ quát kernel thread tương ứng để được xử lý bởi [hệ điều hành](#).

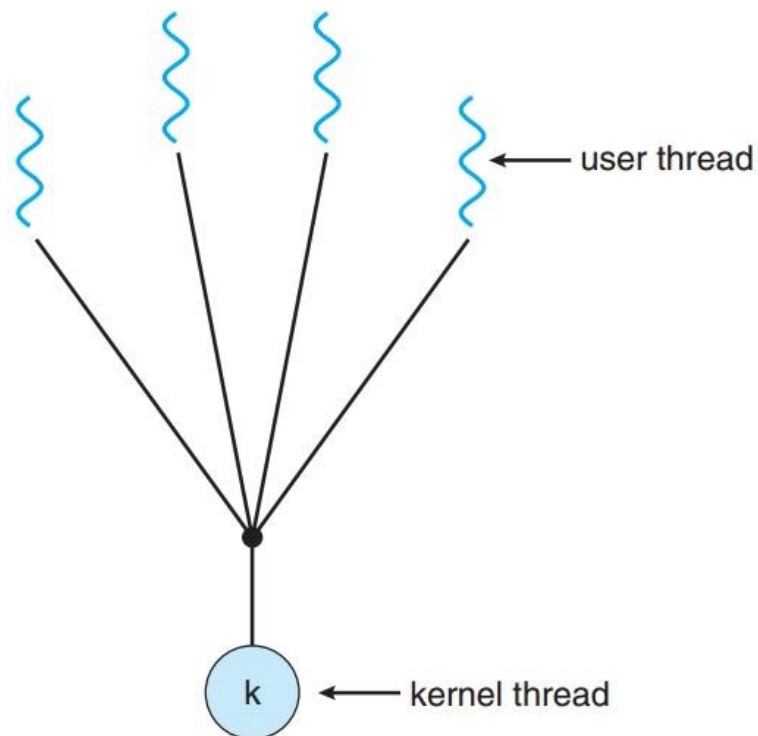


Figure 4.5 Many-to-one model.

mô phỏng MANY-TO-ONE

mô hình many-to-one là mô phỏng phổ thông user threads nối vào một kernel thread. Việc điều hành những luồng này dựa vào thư viện luồng trong không gian các bạn. Tuy nhiên, cả hệ thống sẽ bị chặn nếu 1 luồng nào chậm triển khai thực hành 1 blocking system call. Và cũng vì chỉ 1 luồng được tiếp cận kernel trong một lần, các threads khác không thể chạy song song trong hệ thống đa lõi. Green threads – tên gọi của 1 thư viện luồng của hệ thống Solaris và đã được sử dụng trong các versions cũ của Java – sử dụng mô hình many-to-one. Không sở hữu quá đa dạng hệ thống dùng mô hình này vì nó không sử dụng được lợi thế của multiple processing cores.

mô hình ONE-TO-ONE

mô phỏng one-to-one là mô phỏng 1-1, một user thread kết nối mang 1 kernel thread. Mô phỏng 1-1 này đảm bảo được tính liên tiếp vì ví như 1 thread bị block thì 1 thread khác sẽ vẫn kết nối được sở hữu kernel. Nó cũng đảm bảo được phổ quát luồng sở hữu thể hoạt động cùng một khi trong bộ đa xử lý. Khuyết điểm duy nhất của mô phỏng này là nó đòi hỏi khi một user thread hoạt động thì 1 kernel thread phải được kích hoạt theo. Và vì quá phổ biến kernel threads sẽ gây nên sự quá chuyên chở trong vận dụng, những app tiêu dùng model này đều dùng số lượng thread được tạo ra trong hệ thống. Linux và Windows là những hệ quản lý dùng model one-to-one.

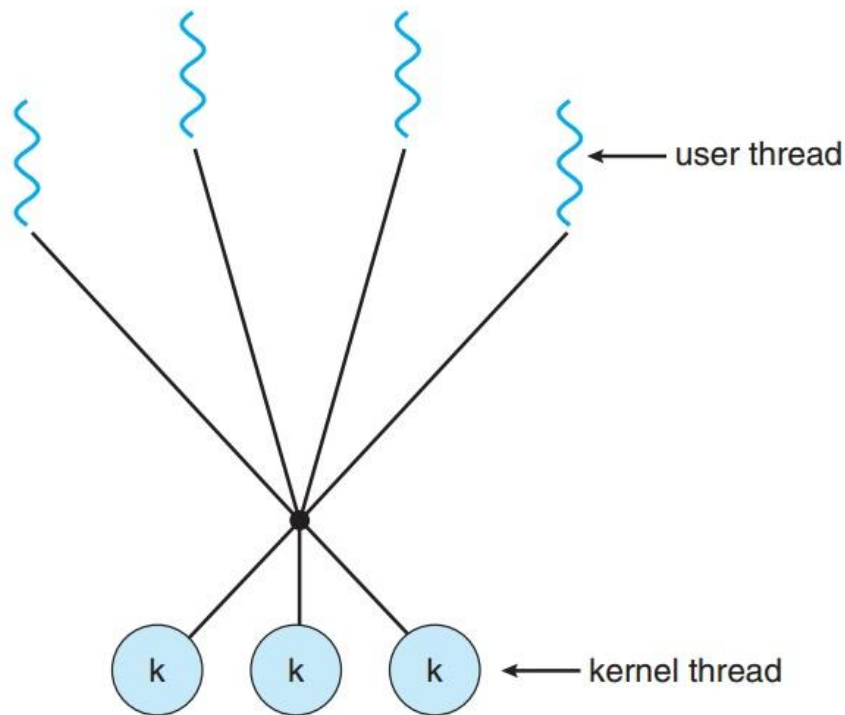


Figure 4.7 Many-to-many model.

mô hình MANY-TO-MANY

mô hình many-to-many chia những user-level threads cho một lượng nhỏ hơn hoặc bằng những kernel threads. Lượng kernel threads này tùy thuộc vào buộc phải của ứng dụng tiêu dùng hoặc bộ máy sử dụng (một ứng dụng thường sử dụng phổ biến kernel threads trên multiprocessors hơn là trên single processor). Mô hình này khác mô phỏng many-to-one ở tính liên tục. Trong mô hình many-to-one, người dùng có thể tạo bao nhiêu user threads tùy thích, nhưng nó không đảm bảo được tính liên tiếp vì 1 kernel chỉ kết nối được sở hữu một user thread, như đã kể ở trên. Mô phỏng one-to-one cho phép tính liên tục cao hơn, ngoài ra số lượng threads được tạo ra rất quan trọng ví như bạn không muốn ứng dụng của mình bị quá

vận tải. Mô phỏng many-to-many sẽ đáp ứng được vấn đề của 2 mô phỏng trước: số lượng user threads tạo ra là tùy thích, và kernel thread tương ứng sở hữu thể chạy song song trong hệ đa xử lý. Và, khi một thread đang thực hành blocking system call, kernel threads tương ứng sở hữu thể chuyển sang một user thread khác và khắc phục user thread chậm tiến độ. 1 Biến thể của many-to-many model là *two-level model*. Model này như là sự hài hòa giữa many-to-many model và one-to-one model, vì nó vừa chia những user-level threads cho một lượng nhỏ hơn hoặc bằng các kernel threads tương ứng, và vừa cho phép một user thread kết nối riêng với 1 kernel thread. Trước version Solaris 9, hệ quản lý Solaris đã dùng *two-level model*. tuy nhiên, từ version 9 trở đi, Solaris sử dụng one-to-one model.

HYPERTHREADING LÀ GÌ

Hyperthreading là khi CPU có khả năng cho 1 core đơn thực thi rộng rãi hơn một luồng cộng 1 khi. Hyperthreading với khả năng nâng cao mức độ xử lý/ tạo đa dạng luồng hơn cho phổ thông core, nhưng chẳng hề toàn bộ core.. Tùy thuộc vào 1 nhiệm vụ đang làm cho, hyperthreading sở hữu thể giúp đem đến hiệu năng khác nhau trong từng core khác nhau, nhưng thỉnh thoảng tổng thể mang thể huy hiệu năng.

mang công nghệ hiện giờ, 1 core mang thể sản sinh ra hai threads, chậm triển khai là lý do với phổ thông CPU đời mới, ví dụ: [ryzen 5 1600](#) sở hữu số core, thread tuần tự là 6, 12. Dù thế, cũng với các con CPU đời mới, ví dụ: [Intel Core i5 7400](#) (Kaby Lake), lại chỉ sở hữu số core, thread lần lượt là 4, 4. Cho nên, tùy thuộc vào công nghệ sử dụng của từng hãng, cho từng con CPU mà số core, số thread nó lại khác nhau.

STREAM HUB - CONTACT INFORMATION

- *Địa chỉ:* 33 Đặng Thai Mai, Phường 7, Quận Phú Nhuận, TPHCM
- *Website:*
 - <http://stream-hub.com/>
 - <http://stream-hub.com/he-dieu-hanh>
- *Facebook:* <https://www.facebook.com/streamhubvn/>
- *Google Site:* <https://sites.google.com/site/streamhub4u/>
- *Google Folder:* <https://goo.gl/yfnfWj>

Chúng tôi luôn cố gắng đưa khái niệm về máy tính trở nên gần gũi với các bạn hơn bao giờ hết và cập nhật những xu hướng tại [Stream Hub](#)

CÁC BÀI VIẾT LIÊN QUAN

[hệ điều hành](#)

[operating system là gì](#)

[tên các hệ điều hành máy tính](#)

[cac he dieu hanh](#)

[giáo trình hệ điều hành](#)

[một số hệ điều hành thông dụng](#)

[các thành phần của hệ điều hành](#)

[hệ điều hành windows](#)

[khái niệm hệ điều hành](#)

[chức năng của cpu](#)

[cpu dien thoai la gi](#)

[cpu nằm ở phần nào của máy tính](#)

[cpu làm những công việc chủ yếu nào](#)

[các loại cpu](#)

[cấu tạo cpu](#)

[những ngộ nhân về ổ cứng ssd](#)

[ổ cứng sata là gì](#)

[đĩa cứng là bộ nhớ trong hay ngoài](#)

[ổ cứng hdd là gì](#)

[các loại ổ cứng ssd](#)

[hệ điều hành free dos là gì](#)

[hệ điều hành free dos](#)

[hệ điều hành được lưu trữ ở đâu](#)

[hệ điều hành là phần mềm gì](#)

[hệ điều hành tiếng anh là gì](#)

[cách xem hệ điều hành máy tính](#)

[hệ điều hành làm những việc gì](#)

[tat chương trình khôi phục win](#)

[hệ điều hành là phần mềm gì](#)

[hệ điều hành tiếng anh](#)

[o cứng hdd là gì](#)

[hệ điều hành](#)

[operating system là gì](#)

[card màn hình cho laptop](#)

[làm sao để kiểm tra card màn hình](#)

CÁC LIÊN KẾT NÊN ĐỌC

<https://goo.gl/4TBQtB>

<https://goo.gl/xJq71X>

<https://goo.gl/JJmPi3>

<https://goo.gl/8PJVL8>

<https://goo.gl/yfnfWj>

<https://goo.gl/exHHf8>

<https://goo.gl/3ia7qU>

<https://goo.gl/5h4tgB>

<https://goo.gl/fKKEDR>

<https://goo.gl/DxRYXL>

<https://goo.gl/7kFFwC>

<https://goo.gl/N1sLov>

<https://goo.gl/jiFtNv>

<https://goo.gl/N7FYGV>

<https://goo.gl/kk867n>

<https://goo.gl/A98g7D>

<https://goo.gl/JM1D7T>

<https://goo.gl/RUc4fK>

<https://goo.gl/67Azwo>

<https://goo.gl/bZ2grS>

<https://goo.gl/3ar3Rr>

<https://goo.gl/aKSEvQ>

<https://goo.gl/WuLMWE>

<https://goo.gl/Lum7j7>

<https://goo.gl/kobsGq>

<https://goo.gl/qcWHKw>

<https://goo.gl/7LYhs1>

<https://goo.gl/zObPyB>
<https://goo.gl/8cX7mQ>
<https://goo.gl/s8Cscm>
<https://goo.gl/UxrU7a>
<https://goo.gl/cWFvRU>
<https://goo.gl/3yaN4S>
<https://goo.gl/fhfXsn>
<https://goo.gl/Zujo9w>